means for passing to a second operand of said graphical operator a modified second region representation in accordance with a second predetermined modification rule for said operator;

means for returning to the graphical operator a third region representation of regions obscured by a sub-tree associated with the second operand; and

means for determining, in accordance with a set rule for said graphical operator, a final region representation to be returned to the parent to form <u>an optimized expression tree in which said final region representation substantially represents an unobscured portion of first region represented at the parent node</u>.

REMARKS

Claims 1-32 are now presented for examination.

Claims 1-3, 6, 8, 11, 17-19, 22, 24 and 27 have been amended to define still more clearly what Applicant regards as his invention.

Claims 1, 8, 11, 17, 24 and 27 are the independent claims.

Claims 1-32 were rejected under 35 USC §103(a) as being unpatentable over "Near Real-Time CSG Rendering Using

10

Tree Normalisation and Geometric Pruning" (Goldfeather) in view

of Australian patent document AU 9523362 A (Politis).

Independent Claim 1, as amended, is directed to a

method of optimizing an expression tree. The expression tree

is used for compositing an image and comprises at least three

nodes, each node of the tree being at least either a graphical

element or a graphical operator. The method comprises, for at

least one node in the tree, a step of comparing a first region

of the node to a second region derived from at least one other

node anywhere in the expression tree. The method also

comprises a step of determining if the first region is totally

or partially obscured by the second region. Finally, the

method comprises a step of modifying the expression tree in the

event that the first region is at least partially or totally

obscured by the second region, to form an optimized expression

tree in which an optimized part of the expression tree

substantially represents unobscured portions of the first

region (see the description at page 9, lines 15 and 16 of the

present specification).

Goldfeather teaches the use of a set of algorithms

for efficiently rendering a CSG-defined object directly into a

frame buffer without first converting to a boundary

representation. The algorithm first converts the CSG tree to a

11

b.

normalised form but is analogous to the sum-of-products form for a Boolean switching function. The reason that Goldfeather converts Boolean expressions into this form is that a CSG expression written as a sum of products can be rendered using two image (i.e., Z and colour) buffer pairs (see disclosure of Goldfeather at page 21). The normalization algorithm taught by Goldfeather uses eight basic set equivalents (see Fig. 2 of Goldfeather) to reduce a CSG tree to normal form. The normalization algorithm matches a tree structure represented at each node to one of the eight basic set equivalences and replaces the tree structure at the particular node with a corresponding set equivalence. Goldfeather acknowledges that the normalization algorithm can create a combinational explosion in the number of nodes (see disclosure of Goldfeather at page 21).

Goldfeather further teaches matching a tree structure represented at each node to one of eight basic set equivalences (see disclosure at page 21, column 2, last paragraph to page 22, column 1 of Goldfeather). Applicant submits that this does not teach or suggest "comparing a first region of the node to a second region derived from at least one other node anywhere in said expression tree", as now recited in Claim 1. After a detailed review of Goldfeather, Applicant only finds a general

12

reference to ignoring terms of the CSG tree that contained non-intersecting (_i.e._, disjunctive) primitives, at page 24 of Goldfeather. At page 24, Goldfeather teaches that the "idea behind bounding-box pruning is to calculate a few numbers that bound the region occupied by a CSG object", and "if the bounding box of an object to be subtracted does not intersect the bounding box of an object from which it will be subtracted, the subtraction need not be computed". However, Applicant respectfully submits that this is very different from the occlusion method as now claimed in the present invention, as will be further explained below.

Applicant respectfully submits that Goldfeather, in general and specifically at pages 21 to 23, does not teach or suggest the particular feature of the present invention of determining if the first region is totally or partially obscured by the second region, as now recited in Claim 1. As described at page 8, line 38 to page 9, line 2 of the present application, a region is compared to the region represented at a node to determine if the region represented by the node is obscured.

In contrast, Goldfeather determines if a tree structure represented at a node matches one of eight basic set equivalences in order to normalize the tree structure. As

13

acknowledged by <u>Goldfeather</u> at page 21, column 1, normalizing the tree structure is analogous to converting a Boolean expression to a sum-of products form. Applicant respectfully submits that <u>Goldfeather</u> does not teach or suggest "determining if said first region is totally or partially obscured by said second region", as now recited in Claim 1. As discussed above, <u>Goldfeather</u> does discuss ignoring terms of a CSG tree that contained disjunctive primitive. However, <u>Goldfeather</u> does not teach or suggest determining if the region represented by a node is totally or partially obscured by one or more other regions, as recited in Claim 1.

Applicant respectfully submits that <u>Goldfeather</u>, in general and specifically at pages 21 to 23 and at Fig. 4, does not teach or suggest modifying the expression tree in the event that the first region is at least partially or totally obscured by the second region, to form an optimized expression tree in which an optimized part of the expression tree substantially represents unobscured portions of the first region, as recited in Claim 1.

As described at page 9, lines 13 to 20 of the present application, if a node is partly obscured by one or more regions represented by other nodes in the expression tree, a clipping operator is applied to the node (ie. the tree is

14

b

modified) in such a way that when executing a compositing operator, substantially unobscured regions of the image represented at the node are in the resultant composite of the region of the node.

In contrast, as discussed above, <u>Goldfeather</u> does not teach or suggest determining if a first region is totally or partially obscured by a second region, and modifying an expression tree in the event that the first region is at least partially or totally obscured by the second region, to form <u>an optimized expression tree in which an optimised part of the expression tree substantially represents unobscured portions of the first region</u>, as now recited in Claim 1. In this connection, <u>Goldfeather</u> states that a CSG expression written as a sum of products can be rendered using two image (i.e. Z and colour) buffer pairs, as discussed above. The use of a Z-buffer is not an optimization of a compositing expression as described and claimed in the present invention. With a Z-buffer, two graphical primitives whose two-dimensional projections overlap are both rendered, but only the front most object finally appears. There is no attempt to reduce the amount of rendering by eliminating or clipping the occluded primitive.

15

Fig. 4 of <u>Goldfeather</u> shows an example of an image rendered in stages in accordance with the method of <u>Goldfeather</u>. The two images at the bottom of Fig. 4 are labelled "front (C)∩D" and "Final Image". The image labeled "front (C)∩D" shows a section of the image rendered for the right-hand side node (i.e. (C)∩D of the expression shown in Fig. 4). The image labeled "Final Image" shows the image as it would be finally rendered. As can be seen from Fig. 4, the image labeled "front(C)∩D" is overwritten when the final image is rendered. Therefore, the compositing expression for a node, modified in accordance with the method of <u>Goldfeather</u>, does not represent an unobscured portion of the region at the node. This particular feature as recited in Claim 1 is not taught or suggested by <u>Goldfeather</u>.

<u>Politis</u> is directed to a system method and language for compositing or creating images. The images typically comprise a plurality of graphical elements each including color and opacity information. The system utilizes operators having the graphical elements and operands in which the operators combine the operands according to a function defined by the operators, the color information, and the opacity information, to produce new graphical elements. One part of the system includes interpreting the language of parsing and executing a

16

sequence of statements and forming an expression tree the nodes
of which comprise a graphical element. Instructions are then
derived from the tree. Another part permits the compositing of
opaque locating active areas of graphical elements from the
nodes.

Politis is cited in the Office Action as remedying
the deficiency of Goldfeather not teaching that the compared
region is of a "representation data structure".

Applicant respectively submits that the addition of
Politis to the teachings of Goldfeather as set out in the
Office Action does not teach or suggest the newly recited
features in Claim 1. A combination of the two references would
produce a method of rendering an image represented by an
expression tree whereby algorithms would be applied to the tree
in order to directly render the objects of the tree into a
frame buffer. The algorithm of Goldfeather would be used to
match the tree structure represented at each node of the tree
to eight basic set equivalence and replace the tree structure
at the particular node with a corresponding set equivalent.
The bounding box methods taught by Politis may then be used for
locating active areas of the graphical elements at the nodes,
in order to keep the node growth under control. However, the
combination of Politis and Goldfeather does not teach or

17

suggest comparing a first region of the node to a second region derived from at least one other node <u>anywhere</u> in the expression tree, as now recited in Claim 1. Further, the combination of <u>Politis</u> and <u>Goldfeather</u> does not teach or suggest determining if the first region is totally or partially obscured by the second region, as now recited in Claim 1. Still further, the combination of <u>Politis</u> and <u>Goldfeather</u> does not teach or suggest modifying the expression tree in the event that the first region is at least partially of totally obscured by the second region, to form an optimized expression tree in which an optimized part of the expression tree substantially represents unobscured portions of the first region, as now recited in Claim 1. For at least those reasons, Claim 1 is considered patentable over <u>Goldfeather</u> and <u>Politis</u> taken alone or in any combination.

Amended independent Claim 8 recites features similar to those recited in Claim 1, and is thought to be patentable for the same reasons. Further, Claims 17 and 24 recite features similar to those recited in Claims 1 to 8, respectively, and are thought to be patentable for the same reasons.

Amended independent Claim 11 is directed to a method of optimising an expression tree for compositing an image, the expression tree comprising a plurality of nodes, each node comprising at least either a graphical element or a graphical operator and having a region of the image represented by the node. The method comprises a step of returning to the graphical operator a second region representation of regions obscured by a sub-tree associated with the first operand. The method also comprises a step of determining, in accordance with a set rule for the graphical operator, a final region representation to be returned to the parent node to form an optimized expression tree in which an optimized part of the expression tree substantially represents unobscured portions of the first region.

Applicant respectfully submits that neither Goldfeather nor Politis teaches or suggests returning to the graphical operator a second region representation of regions obscured by a sub-tree associated with the first operand, as now recited in Claim 11. Further, neither Goldfeather nor Politis nor any combination thereof, suggests determining, in accordance with a set rule for the graphical operator, a final region representation to be returned to the parent node to form an optimized expression tree in which an optimized part of the

19

expression tree substantially represents unobscured portions of the first region, as now recited in Claim 11.

At page 24, <u>Goldfeather</u> includes a general reference to ignoring terms of the CSG tree that contain non-intersecting primitives by using bounding boxes. At page 19, lines 15 to 30, <u>Politis</u> teaches reducing the size of a bounding box to detect when a first child node of a parent node is obscured by a second chid node of the parent node. Further, at this reference, <u>Politis</u> teach eliminating the bounding box of a first child node depending on whether it is totally obscured by that of a second child node which is directly linked with the first child node. Therefore, there is no teaching or suggestion of returning to the graphical operator a second region representation of regions obscured by a sub-tree associated with the first operand as now recited in Claim 11. This particular feature is described at page 12, lines 36 to 37 of the present application, and provides a different optimization from that taught by <u>Politis</u>. The bounding box taught in <u>Politis</u> extends over the combined areas of only two child nodes having the same parent node. Applicant respectfully submits, that this is different from the region which is returned to the graphical operations and represents regions obscured by a sub-tree associated with a first operand,

20

as now recited in Claim 11. For at least those reasons, Claim 11 is considered patentable over <u>Goldfeather</u> and <u>Politis</u> taken separately or together.

Amended independent Claim 27 recites features similar to those recited in Claim 11, and is thought to be patentable for the same reasons.

A review of the other art of record has failed to reveal anything which, in Applicant's opinion, would remedy the deficiencies of the art discussed above as references against the independent claims herein. Those claims are therefore believed patentable over the art of record.

The other claims in this application are each dependent from one or another of the independent claims discussed above, and are therefore believed patentable for the same reasons. Since each dependent claim is also deemed to define an additional aspect of the invention, however, the individual reconsideration of the patentablility of each of its own merits is respectfully requested.

In view of the foregoing amendments and remarks, Applicant respectfully requests favorable reconsideration and early passage to issue of the present application.

Applicant's undersigned attorney may be reached in our New York office by telephone at (212) 218-2100. All

21

correspondence should continue to be directed to our below

listed address.

Respectfully submitted,

Jock M. Arnold

Attorney for Applicant

Registration No. 25,823

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3801
Facsimile: (212) 218-2200

NY_MAIN 90493 v 1

22